

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Dalam penelitian ini akan digunakan tinjauan pustaka yang dapat mendukung penelitian, berikut ini merupakan tinjauan pustaka yang diambil :

Tabel 2.1 Tinjauan Pustaka

Nomor Literatur	Penulis	Tahun	Judul
Literatur 1	Jan Bayer dan Jan Faigl	2019	On Autonomous Spatial Exploration with Small Hexapod Walking Robot using Tracking Camera Intel RealSense T265
Literatur 2	Ankit Agarwal, Jacob R, Crouse dan Eric N. Johnson.	2020	Evaluation of a Commercially Available Autonomous Visual Inertial Odometry Solution for Indoor Navigation
Literatur 3	Shinya Kawabata, Jae Hoon Lee dan Shingo Okamoto.	2019	Obstacle Avoidance Navigation Using Horizontal Movement for a Drone Flying in Indoor Environment

Literatur 4	Jan Bayer dan Jan Faigl	2020	Vision-based Localization for Multi-rotor Aerial Vehicle in Outdoor Scenarios
Literatur 5	Diah Liani, Ade Silvia, dan Lindawatis	2017	Sistem Navigasi Pada Mobile Robot Dalam Penentuan Arah dan Pemetaan Posisi

### 2.1.1 Tinjauan Pustaka Literatur 1

Pada penelitian ini membahas tentang penerapan kamera Intel Realsense T265 dan Kamera RGB-D Intel Realsense D435 dalam navigasi Robot Mini Hexapod. Pada penelitian ini, Intel Realsense D435 digunakan untuk mengumpulkan *Cloud Point* 3D dari lingkungan sekitar robot dan dibantu dengan Intel Realsense T265 untuk mendapatkan estimasi pose robot menggunakan Visual SLAM (*Simultaneous Localization and Mapping*) yang memanfaatkan kamera *fisheye* stereo dan *Unit Pengukuran Inersia* (IMU) (Jan Bayer & Jan Faigl, 2019).

### 2.1.2 Tinjauan Pustaka Literatur 2

Pada penelitian ini membahas tentang kemampuan Intel Realsense T265 serta mengevaluasi keajaiban *Extended Kalman Filter* (EKF) dalam menghaluskan noise sensor dan tindak lanjut untuk diperintahkan dalam penerbangan di lokasi tertutup. Kesalahan yang disebabkan oleh keterbatasan inheren dari titik fitur dan IMU menghasilkan lonjakan berupa akselerasi secara tiba-tiba atau perubahan arah yang cepat berada di luar kemampuan Intel Realsense T265. Selain itu, faktor pencahayaan yang tidak konsisten dan getaran menyebabkan perangkat lebih mudah kehilangan jejak poin fitur serta akurasi ikut menghilang.

Pengujian yang dilakukan menggunakan posisi awal dan tersaring oleh data pos dari T265 kemudian data ini dibandingkan dengan hasil video sistem penangkapan untuk memvalidasi EKF untuk menyaring keluaran T265 yang bernoise (Ankit Agarwal, et al., 2020).

### **2.1.3 Tinjauan Pustaka Literatur 3**

Pada penelitian ini menggunakan kamera Intel Realsense T265 yang dikombinasikan dengan *Robotic Operation System* (ROS) untuk melakukan estimasi posisi dan penghindaran tabrakan. Dalam implementasinya, peneliti memanfaatkan *Visual Inertial Odometry* (VIO) sebagai algoritma estimasi, dimana posisi dan postur tubuh dihitung dengan nilai terukur dari percepatan dan kecepatan sudut dari sensor IMU. Untuk melakukan penghindaran, peneliti menggunakan system eksternal seperti pelacak gerak Optitrack (Shinya Kawabata, et al., 2019).

### **2.1.4 Tinjauan Pustaka Literatur 4**

Pada penelitian ini menggabungkan dua kamera Intel Realsense T265 dan D435. Hasil dari kedua kamera dan semua sensor dikumpulkan menggunakan middleware *Robotics Operating System* (ROS). Lokalisasi dari kamera T265 dan D435 ditangkap langsung selama percobaan. Lokaliasasi yang disediakan oleh ORB-SLAM2 dihasilkan dengan memproses data RGB-D dari dataset rosbag. Sistem lokalisasi yang diperiksa Intel Realsense T265 menyediakan hasil yang petitif dalam scenario penyebaran luar ruangan yang realistis. Di setengah dari percobaan eksperimental, T265 lebih baik dari pada ORB-SLAM2. Pengaruh kecepatan dan sudut tinggi tidak diamati untuk T265, yang membuatnya lebih unggul dari ORB-SLAM2 dimana pengaruh tersebut

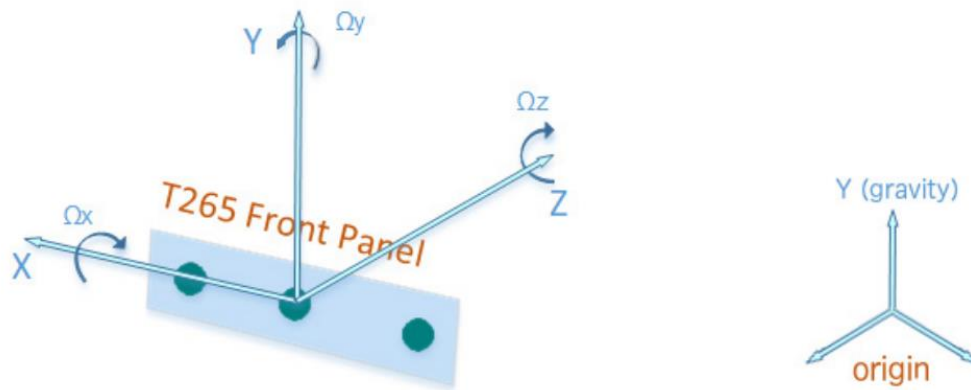
dibutuhkan. Efek ketinggian penerbangan pada T265 hanya diamati pada ketinggian diatas 15 meter, lebih dari itu akan mengalami gerakan yang tidak normal (Jan Bayer, et al., 2020).

### **2.1.5 Tinjauan Pustaka Literatur 5**

Pada penelitian ini dirancang mobile robot dengan kemampuan mengatur arah gerak menuju arah yang dituju menggunakan compass sebagai penentu arah. Untuk melakukan gerak arah yang dituju menggunakan metode waypoint. Hasil dari penelitian ini Modul Compass memiliki selisih Rata-rata sebesar  $4^\circ$  setiap interval 5 detik dan gerak rotasi dapat mencapai target compass yang ditargetkan dengan selisih terbesar antara target tercapainya compass sebesar  $+3^\circ$ . Sehingga pembacaan nilai kompas juga dipengaruhi dengan benda yang ada di sekitarnya sehingga pergerakan robot tidak sesuai ketika dijalankan di dalam area indoor dan akan lebih baik jika dilakukan di area outdoor (Liani, et al., 2017).

## **2.2 Intel Realsense T265**

Intel Realsense T265 merupakan perangkat *Visual Inertial Odometry* (VIO) yang tersedia secara resmi, dikembangkan dan disajikan dengan *Extended Kalman Filter* (EKF). VIO dalam T265 menggabungkan kekuatan navigasi berbasis visi dan pengukuran IMU secara efektif. Selain IMU, Intel Realsense T265 berisi kamera *fisheye* stereo terkalibrasi yang menyediakan pelacak titik fitur akurat serta koreksi skala.



Gambar 2.1 Kerangka Koordinat Lokal T265

Dalam konfigurasinya T265 menyediakan pustaka Intel Realsense SDK 2.0 menyediakan API dimana data dari T265 dapat diinisialisasi dan diakses. Perpustakaan Librealisense memungkinkan integrasi data streaming T265 tanpa repot ke basis kode dan pipeline ini diintegrasikan ke dalam control perangkat lunak penerbangan yang berjalan diatas Robot. Secara efektif, perangkat lunak control penerbangan mengakses sensor update loop yang menyediakan data real-time ke navigasi sistem. Di dalam loop pembaruan ada sensor antar muka untuk berbagai sensor yang mungkin dimiliki robot, termasuk GPS, IMU, LiDar dan lainnya. T265 terintegrasi ke dalam perangkat lunak dan mengirimkan perkiraan pose yang diukur ke sistem navigasi. Sistem navigasinya sendiri dirancang untuk menjalankan loop sensor pada frekuensi yang lebih tinggi dari pada tingkat pembaruan 200 Hz dari T265. Sebelum berpose pengukuran dikirim ke sistem navigasi, namun itu harus diubah menjadi kerangka koordinat yang dapat digunakan.

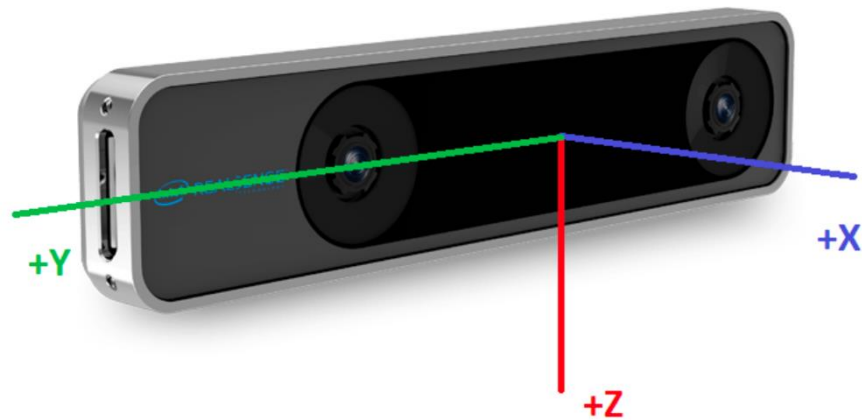
T265 menggunakan standar VR “Kanan, atas, belakang” sistem koordinat membantu integrasi dengan sistem serupa. Saat pelacakan dimulai, koordinat global diinisialisasi dengan sumbu y selalu mengarah ke berlawanan arah dari

perepatan gravitasi yang diukur vector, sumbu x dan y diinisialisasi berdasarkan orientasi kamera. T265 kemudian memberikan pose relative ke bingkai global.

Kerangka koordinat umum untuk robot udara adalah Sistem “Depan, kanan, bawah”. Output pose dari T265 harus dikonversi dari bingkai aslinya ke kerangka robot, sambil juga memperimbangkan erasi sudut pemasangan khusus perangkat ke kendaraan. Selain itu, kerangka koordinat kendaraan adalah berpusat di *Center of Gravity* (CG) robot, yang secara umum tidak betepatan dengan pusat koordinat T265. Jadi, baik rotasi maupun koreksi terjemahan harus diterapkan untuk penggunaan yang tepat dalam sistem navigasi robot.

Dalam uraian berikut, notasi akan digunakan untuk mendeskripsikan bingkai koordinat dan transformasi:

- Sebuah subskrip kanan akan digunakan untuk menunjukan dua titik untuk posisi atau titik, bingkai untuk kecepatan dan percepatan. Sebuah garis miring “/” dalam arti subskrip “*with respect to*”.
- Superskrip yang tepat akan menunjukan sistem koordinat dimana property pose akan diekspresikan.
- Untuk matriks arah kosinus (DCM) yang pertama dan istilah kedua dalam subskrip menunjukan bingkai target dan bingkai awal.



Gambar 2.2 Bingkai Lokal Baru Setelah Transformasi Koordinat

Misalnya, T265 menyediakan data dalam bingkai sensor ( $s$ ) sehubungan dengan bumi ( $E$ ), koordinat local diekspresikan ( $L$ ). vector posisi mentah dari sensor akan dilambangkan sebagai  $\vec{P}_{s/E}^L$ . Pengukuran dari T265 harus dikonversi ke kerangka bodi robot ( $b$ ) sehubungan dengan bumi, dinyatakan dalam koordinat lokal. Jadi, vektor posisi transformasi yang harus disediakan adalah  $\vec{P}_{s/E}^L$ .

1) Koreksi posisi & rotasi

Untuk posisi solusinya berikut ini diketahui:

- a) Vektor posisi dari T265 terhadap bumi diekspresikan dalam bingkai

$$\text{VR local, } \vec{P}_{s/E}^{V R}$$

- b) Vektor dari CG robot ke pelacakan bagian tengah T265,  $\vec{r}_{mount/CG}$

- c) Rotasi quaternion dari sensor,  $q_s$

- d) Offset rotasi sensor relative terhadap robot karena mountingnya,

$$q_{mount}$$

Sebagai Langkah awal, transformasi koordinat menggunakan DCM  $C_{L/VR}$  akan diterapkan ke keluaran posisi mentah dari T265 untuk mengubah bingkai koordinat VR menjadi yang baru yang di sebut bingkai sensor. Ini baru bingkai s dihitung menggunakan persamaan. 1, harus mengungkapkan posisi dari kordinat FRD, seperti pada gambar.

$$\vec{P}_{s/E}^L = C_{L/VR} \vec{P}_{s/E}^{VR}$$

Dimana

$$C_{L/VR} = \begin{bmatrix} 0 & 0 & -0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Kemudian, solusi posisi diubah dari bingkai local ke rangka tubuh:

$$\vec{P}_{s/E}^B = C_{B/L} \vec{P}_{s/E}^L$$

Dimana  $C_{B/L}$  adalah matriks kosinus arah:

$$C_{B/L} = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}$$

Dan  $q$  adalah total putaran angka empat dari sensor :

$$q = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k} + q_s + q_{mount}$$

Kemudian, solusi posisi pada rangka bodi diterjemahkan untuk memberikan posisi CG kendaraan :

$$\vec{P}_{b/E}^B = \vec{P}_{s/E}^B - \vec{r}_{mount/CG}$$

Terakhir, posisi CG kendaraan doiubah dari bingkai tubuh kembali ke bingkai local:

$$\vec{P}_{b/E}^L = C_{L/B} \vec{P}_{b/E}^B$$



$$= (C_{L/B})^T \vec{P}_{b/E}^B$$

## 2) Koreksi kecepatan

Untuk solusi kecepatan, berikut ini diketahui (tambahan untuk semua parameter yang ditentukan sebelumnya):

- a) Vektor kecepatan dari T265 terhadap bumi diekspresikan dalam bingkai,  $\vec{v}_{s/E}^{VR}$
- b) Tingkat sudut sensor terhadap bumi diekspresikan dalam bingkai sensor,  $w_{s/E}^S$

Seperti halnya koreksi posisi, Langkah awal akan dilakukan untuk mengubah vector kecepatan dalam bingkai koordinat VR dari T265 ke bingkai FRD menggunakan koordinat matriks yang sama.

$$\vec{v}_{s/E}^L = C_{L/VR} \vec{v}_{s/E}^{VR}$$

Vektor kecepatan diubah dari bingkai local menjadi kerangka tubuh menggunakan matriks arah cosinus.

$$\vec{v}_{s/E}^B = C_{B/L} \vec{v}_{s/E}^L$$

Persamaan kecepatan total ditetapkan untuk menentukan kecepatan CG kendaraan.

$$\vec{v}_{b/E}^B = \vec{v}_{s/E}^B - \vec{\omega}_{b/E}^B \times \vec{r}_{mount/CG}$$

Dimana

$$\vec{\omega}_{b/E}^B = \vec{\omega}_{s/E}^S C_{B/S}$$

dan  $C_{B/S}$  ditentukan dari orientasi sudut Euler dari dudukan T265 ( $\Phi$  adalah sudut pitch,  $\theta$  adalah sudut roll, dan  $\Psi$  adalah sudut yaw).

$$C_{B/S} = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

Akhirnya, kecepatan CG kendaraan diubah dari bingkai tubuh Kembali kebingkai lokal.

$$\begin{aligned} \vec{v}_{b/E}^L &= C_{L/B} \vec{v}_{b/E}^B \\ &= (C_{B/L})^T \vec{v}_{b/E}^B \end{aligned}$$

### 3) Koreksi akselerasi

Untuk solusi akselerasi atau percepatan, diantara lain meliputi (selain semua parameter yang ditentukan sebelumnya).

- a) Vektor percepatan T265 terhadap bumi di ekspresikan dalam bingkai local,  $\vec{a}_{s/E}^{VR}$
- b) Percepatan sudut sensor terhadap bumi diekspresikan dalam bingkai sensor,  $\vec{a}_{s/E}^S$

Seperti sebelumnya, vektor percepatan dalam bingkai VR adalah yang pertama dikonversi ke bingkai FRD.

$$\vec{a}_{s/E}^L = C_{L/VR} \vec{a}_{s/E}^{VR}$$

Vektor percepatan kemudian diubah menjadi kerangka tubuh menggunakan DCM.

$$\vec{a}_{s/E}^B = C_{B/L} \vec{a}_{s/E}^L$$

Persamaan percepatan total diterapkan untuk menentukan akselerasi CG kendaraan.

$$\vec{a}_{b/E}^B = \vec{a}_{s/E}^B - (\vec{a}_{b/E}^B \times \vec{r}_{mount/CG})$$

$$- \left( \vec{\omega}_{b/E}^B \times \left( \vec{\omega}_{b/E}^B \times \vec{r}_{mount/CG} \right) \right)$$

Dimana

$$\vec{\alpha}_{b/E}^B = \vec{\alpha}_{s/E}^S C_{B/S}$$

Akhirnya, percepatan CG robot diubah dari bingkai tubuh Kembali ke bingkai lokal.

$$\begin{aligned} \vec{\alpha}_{b/E}^L &= C_{L/B} \vec{\alpha}_{b/E}^B \\ &= C_{B/L}^T \vec{\alpha}_{b/E}^B \end{aligned}$$

### 2.3 Localization

Localization adalah sistem independent yang memberikan estimasi pose terbaik untuk membantu kesalahan eksplorasi robot serta kemampuan dari robot untuk bergerak menuju tempat tujuan (Jan Bayer, et al., 2020).

### 2.4 Mavlink

Micro Air Vehicle Link (MAVLink) adalah protokol komunikasi untuk sistem tak berawak (mis., Drone, robot). Ini menentukan serangkaian pesan komprehensif yang dipertukarkan antara sistem tak berawak dan *ground station*. Protokol ini digunakan dalam sistem *flight control* utama, terutama ArduPilot dan PX4, dan menyediakan fitur untuk memantau dan mengendalikan misi sistem tak berawak (Koub^aa, et al., 2019).